

**AMENDMENTS TO THE CLAIMS:**

The following listing of claims supersedes all prior versions and listings of claims in this application:

1. (Currently Amended) A computer system comprising:  
a plurality of components that can be initialized, each component being operable to perform at least one task when not being initialized, but disrupting performance of said at least one task when being initialized,  
wherein each component is configured to produce status data from which the current level of need for that component to be initialized can be inferred, the status data representing a predetermined current level of need for that component to be initialized, and  
wherein at least one of said components is itself also configured to:  
receive status data from other components;  
make a comparison using its own status data and the status data received from respective other components;  
in dependence on the comparison, select one or more of said components, possibly itself, for initialization; and  
issue initialization instructions to the selected component(s).

2. (Previously Presented) A computer system as claimed in claim 1, wherein each component configured to make a comparison using status data is configured to use its own status data in addition to received status data when making the comparison.

3. (Previously Presented) A computer system as claimed in claim 1, wherein the components are software components, and wherein the system includes at least one computer device on which, in use, the software components are run.

4. (Previously Presented) A computer system as claimed in claim 1, wherein the status data is in the form of an initialization parameter.

5. (Previously Presented) A computer system as claimed in claim 4, wherein:  
each component is configured to execute an initialization routine when the initialization parameter for that component reaches a respective threshold value,  
the initialization routine including the step of transmitting a request for an initialization parameter to other components.

6. (Previously Presented) A computer system as claimed in claim 5, wherein the initialization routine includes the further steps of:  
receiving initialization parameters from at least some of those other components;

comparing the received initialization parameters with the initialization parameter for that component; and  
in dependence on the comparison, making a self-initialization decision.

7. (Previously Presented) A computer system as claimed in claim 4, wherein:  
each component includes a timer module for registering elapsed time since previous initialization of that component, and

for each component, the initialization parameter is determined at least in part in dependence on the elapsed time registered by the timer module.

8. (Previously Presented) A computer system as claimed in claim 4, wherein each component is configured to produce an initialization parameter that is at least in part dependent on whether the component is performing one of a number of predetermined tasks.

9. (Previously Presented) A computer system as claimed in claim 3, wherein the computer system includes a plurality of interconnected computer devices, each of which is housed in a respective housing, and wherein each device has, in use, a respective software component running thereon.

10. (Original) A computer system as claimed in claim 9, wherein the software components each include a respective operating system module for operating the computer device on which the respective software component is running.

11. (Previously Presented) A computer system as claimed in claim 9, wherein each component is configured to initiate a re-boot routine upon receipt of an initialization instruction, the re-boot routine being configured to re-boot the computer device on which the software component is running.

12. (Original) A computer system as claimed in claim 11, wherein the re-boot routine includes the step of determining if the computer device is performing a predetermined task or one of a number of predetermined tasks, and only to permit the re-booting of the computer device if the computer device is not performing such a task.

13. (Previously Presented) A computer system as claimed in claim 3, wherein the components in use run on a common computer device, under the control of a common operating system.

14. (Previously Presented) A computer system as claimed in claim 3, wherein each component, upon receipt of an initialization instruction, is configured such that the component is killed and subsequently restarted.

15. (Previously Presented) A computer system as claimed in claim 1, including a computer device configured to allocate tasks to the components, such that a task allocated to one component is dependent on the task or tasks being performed by at least some of the other components.

16. (Currently Amended) A method of initializing the components of a computer system, the computer system having a plurality of components that can be initialized, each component being operable to perform at least one task when not being initialized, but disrupting performance of said at least one task when being initialized, each component being configured to produce status data from which the current level of need for that component to be initialized can be inferred, the method comprising at least one of said components performing the steps of:

receiving status data from other of said components;

making a comparison between the status data for all said components including status data received from respective components and its own status data;

in dependence on the comparison, selecting one or more components, possibly itself, for initialization; and  
issuing initialization instructions to the selected component(s).

17-22. (Cancelled)

23. (Currently Amended) A computer system comprising:  
a plurality of components that can be initialized, each component being operable to perform at least one task when not being initialized, but disrupting performance of said at least one task when being initialized,

wherein each component is configured to produce status data from which the current level of need for that component to be initialized can be inferred, and

wherein at least one of said components is configured to:

receive status data from said other components;

make a comparison using its own status data and the status data received from respective said other components;

in dependence on the comparison, select one or more components, possibly itself, for initialization; and

issue initialization instructions to the selected component(s).

24. (Currently Amended) A method of initializing the components of a computer system, each component being operable to perform at least one task when not being initialized, but disrupting performance of said at least one task when being initialized, the method comprising:

using status data from itself and a plurality of other of said components to determine for each said component, the current need for that component to be initialized relative to the need for at least one other of said components to be initialized; and

initializing at least one of the components in dependence on its so determined relative need.

25. (Previously Presented) A computer system as in claim 1, wherein said status data is fault-related.

26. (Previously Presented) A method as in claim 24, wherein said status data is fault-related.